Glenn Procedural Requirements

GLPR 8739.1A

Effective Date: February 6, 2013
Expiration Date: February 6, 2018

COMPLIANCE IS MANDATORY

Responsible Office: Code Q/Safety and Mission Assurance Directorate

Software Assurance

TABLE OF CONTENTS

Change History

Change	Date	Description/Comments
Basic	9/24/2007	Document converted from CLP (GRC-P2.10.2) to GLPR.
A	2/6/2013	Document updated to NPR 7150.2A and changes made to address the Quality Audit, Assessment, and Reviews (QAAR) and Requirement Engineering Design Audits and Assessments (REDAA) findings.

Preface

P.1 Purpose

The purpose of this procedure is to:

(1) Provide Center-level requirements for the development, documentation, and implementation of a software assurance program.

(2) Define software assurance tasks and outline processes to ensure safety, reliability, and quality of all software products.

(3) Implement the requirements of NASA-Standard (STD)-8739.8, Software Assurance Standard.

(4) Support and utilize the independent reporting structure required for Safety and Mission Assurance (SMA) processes.

P.2 Applicability

a. This Glenn Procedural Requirements (GLPR) document applies to class A through D (per NASA Procedural Requirements (NPR) 7150.2 software classification) software, E and safety-critical (per NASA-STD-8739.8) software, and class F through G software developed, acquired, or managed at Glenn Research Center (GRC) during the entire software life cycle, regardless of developmental models. This includes the software tools and simulators created for developing, verifying, or validating software/hardware systems used in safety-critical missions, mission-critical projects, or critical facilities. Firmware shall be treated as software.

b. When open-source, legacy, reused software products, Government-off-the-shelf (GOTS) and commercial-off-the-shelf (COTS) are used as is without any modifications or alterations, then the program/project/subproject/task management is responsible for obtaining verification that such software products received adequate quality assurance. However, if such software is to be used for flight, or if there are any modifications, alterations, additions, parameters, or applications, and bridging software written in order to use them, then this procedure shall be applied. See Guidance E.11 in Appendix E.

c. Tailoring:

(1) Program/project/subproject or task managers, working with software assurance personnel, shall use the software assurance classification assessment in NASA-STD-8739.8, Software Assurance Standard, (Appendix A) to determine the appropriate level of software assurance effort.

(a) The software assurance classification report template in Appendix I (of this document) shall be used to document this assessment.

(b) A reassessment of the software assurance classification shall be performed when there is a major change in the requirements or in the project status from developmental to flight status. See Appendix F for more details on tailoring of software assurance (SA) requirements.

(2) Often Class D assurance activities consist mostly of assuring any contractual agreements meet the needs of the project/program and then performing periodic audits and surveys of the project work to follow up. The level of software assurance effort applied to any class is commensurate with the risk, criticality, complexity, and needed reliability and quality of a project.

(3) If the results of the software assurance classification assessment in NASA-STD-8739.8, (Appendix A) identify the software as Class E (which includes exploratory software) and not safety-critical, then the requirements of this document are not mandatory.

(4) Class F through H software is currently the responsibility of the Chief Information Office, however, for the higher-level information technology or business class systems, if software assurance is requested, those projects would be assured in accordance with the software engineering requirements in NPR 7120.5 they must meet.

d. This directive is applicable to documents developed or revised after the effective date of this GLPR.

e. In this directive, all mandatory actions (i.e., requirements) are denoted by statements containing the term "shall." The term "may" or "can" denotes discretionary privilege or permission; "should" denotes a good practice and is recommended, but not required; "will" denotes expected outcome; and "are" or "is" denotes descriptive material.

f. In this directive, all document citations are assumed to be the latest version, unless otherwise noted.

P.3 Authority

NASA Policy Directive (NPD 2810.1), “NASA Information Security Policy”

P.4 Applicable Documents and Forms

a. NPR 7120.5, “NASA Space Flight Program and Project Management Requirements”

b. NPR 7150.2, “NASA Software Engineering Requirements”

c. NPR 8715.3, “NASA General Safety Program Requirements”

d. Institute of Electrical and Electronics Engineers (IEEE) 610.12, IEEE Standard Glossary of Software Engineering Terminology

e. IEEE 730-2002, “IEEE Standard for Software Quality Assurance Plans”

f. NASA-STD-8719.13, “NASA Software Safety Standard”

g. NASA-STD-8739.8, “NASA Software Assurance Standard”

h. GLPR 7120.5.30 “Space Assurance Requirements”

i. GLPR 7150.1, “GRC Software Engineering Requirements

j. NASA Form (NF) 1707, “Special Approvals and Affirmations of Requisitions”

P.5 Measurement/Verification

The SA manager shall collect SA metrics per project to measure the effectiveness of this procedure and report the metrics data to the GRC Quality Engineering and Assurance Branch.

P.6 Cancellation

This GLPR cancels GLPR 8739.1, Software Assurance, dated September 24, 2007.

/s/
James. M. Free
Director

CHAPTER 1. Introduction

1.1 Introduction

Software assurance (SA) is a multidisciplined function which consists of software quality (software quality assurance, software quality control, and software quality engineering), software safety, software reliability, software verification and validation (V&V), and software independent verification and validation (IV&V), with a common purpose to assure that software products are high quality and operate safely while ensuring mission success. The SA process is the planned and systematic set of activities, performed by many different groups, which ensure conformance of software development processes and products to requirements, standards, and procedures. Thus, this procedure is tied to GLPR 7150.1 and must be used in conjunction with it. Once the project has evaluated for the appropriate software classification, then the SA tasks and level of effort can be assessed and documented within the project plan. In addition, the SA function also serves as a resource for information, advice, analysis, and independent reporting on quality, reliability, and safety of software products. However, assurance is part of every aspect of the development process and as such, is everyone's responsibility as well.

1.2 Records

a. The following tables show the SA documents and records required for each SA classification.

b. While created and maintained by the project, the SA Plan and the Risk Management Plan require SA input and sign off. The SA Plan, whether part of the software management plan or stand-alone, is the agreement between management and SA team of the SA tasks to be performed. Inspection reports, audit reports, software quality and safety procedures, discrepancy reports, and their metrics, may be maintained by SA personnel or kept in program/project/subproject/task files. The location and responsibility for these records and documents are described in the SA Plan.

c. See Appendix E of this document for guidance and examples of these documents and records.

	SA Effort/Prioritization Criteria (taken from NASA-STD-8739.8 table A-3)			
	Full/ High	**Full/ Medium-High**	**Medium/ Medium**	**Minimal/ Low**
Software Project /Sub-project/Task Owned Documents:				
Software Assurance Plan	R	R	R	E
Risk Management Plan	R	R	E	O
Software Assurance Owned Document:				
Software Quality Assurance Procedures	R	R	E	E
Software Safety Procedures	R	E	O	O
Software Assurance Quality Records:				
Software Safety Litmus Test &	R	R	R	E

Software Assurance Classification Assessment Report				
Document Review Inputs	R	R	R	E
Inputs to Reviews and Boards	R	E	O	O
Analysis Results	R	R	O	O
Audit/Assessment Results	R	R	E	E
Discrepancy Reports	R	R	O	O
Inspection Reports	R	E	O	O
Participation in Software Verification and Validation	R	R	E	O
Metrics collection and reporting	R	R	E	E
Recommended preventive measures, and lessons learned	R	R	E	E

NOTE: All of these documents and records may be stand-alone or part of system-level documentation.

Key: E=Evaluate
R=Required
O=Optional

1.3 Quality System

1.3.1 A quality system shall be implemented and maintained by the project or program offices.

1.3.2 The quality system shall incorporate SA functions.

a. Those performing the SA functions, while working with the project engineers and programmers as a team member, shall report findings directly to project, facility, and assurance management as defined in the Product Assurance Plan (PAP). The SA functions should be independent of the Software Engineering and Project Management organization in order to assure effective and objective oversight and insight to the project/program.

b. For safety-critical software, the SA functions related to safety shall be independent and work in coordination with the safety organizations/offices.

c. The SA function shall provide analysis, review, advice, and verification in order to produce a quality system which includes planning and a systematic approach to the evaluation of software processes and products and adherence to agreed upon software procedures, plans, standards, and guidelines.

CHAPTER 2. Responsibilities

2.1 Project, Subproject, and Program Management

2.1.1 The project, subproject, and program management determine the classification and safety criticality of the software and then scope the SA functions for that project, product, or program.

a. For acquired software, the project, subproject, and program management shall ensure completion of NF 1707 for all procurements subject to NPR 7150.2.

b. The responsible GRC office shall also meet the requirements as laid out in the Space Assurance Requirements (GLPR 7120.5.30) for SA activities.

c. If the software is deemed safety-critical, then program/project/subproject management shall follow the NASA Software Safety Standard, NASA-STD-8719.13, and assign a software safety person responsible for implementing NASA-STD-8719.13.

d. The project/program/subproject management shall assure that software engineering, software assurance, systems engineering, and system safety are all coordinated and working together as needed.

2.1.2 The extent to which a project utilizes or creates the function of SA shall be determined by program/project/subproject management with input from the software lead and the SA manager.

a. The exact tasks and analyses to be performed, as well as who has the responsibility to perform them, will be negotiated.

b. The agreed-upon activities shall be documented in a PAP.

2.1.3 Program/project/subproject management shall assure that there are the necessary resources, including trained personnel, appropriate equipment, and software tools to perform the SA function.

2.2 The SMA Directorate

The SMA directorate, if requested on a project, must insure that there are sufficient SA engineers to cover the work on projects and facilities at GRC. In addition, they shall assure that the SA engineers are properly trained in the latest software assurance, safety, management, and development techniques. The time and activities to be supported are negotiated with project management and with the approval of SA recorded in the PAP.

2.3 Acquirer SA Manager

2.3.1 This refers to the person who is appointed to be responsible for directing and managing the acquirer SA program.

2.3.2 The acquirer SA manager shall perform an initial software safety litmus test per Appendix H and SA classification assessment per Appendix I of this procedure.

a. When the software system is determined to be safety-critical, then the SA manager shall ensure that the software product and process comply with the requirements in NASA-STD-8719.13.

b. The acquirer SA manager shall provide surveillance to assure that both the acquirer and provider SA functions are performed according to their specific SA plans and the contract.

c. Specifically, the acquirer SA manager shall ensure that acquirer SA staff performs tasks to provide both insight and oversight over provider's management, assurance, and engineering plans and processes according to the acquirer SA plan.

d. The acquirer SA manager shall ensure that SA processes and planning are in place for operation and maintenance of software developed or acquired by NASA, including periodic audits and software configuration management.

e. The acquirer SA manager shall assure that a software retirement plan is prepared, approved, and executed, including archival or disposal of SA records and documents.

2.4 Provider SA Manager

2.4.1 This refers to the person who is appointed to be responsible for directing and managing the provider SA program.

2.4.2 The provider SA manager shall:

a. Conduct and document periodic reviews of the SA process, periodic reviews, audits, and assessments of the development process and products.

b. Prepare SA status reports in accordance to NASA-STD-8739.8.

c. Flow down the requirements of the NASA-STD-8739.8 to any subcontractor and assure that the subcontractors satisfy these requirements.

d. Perform a detailed software safety criticality assessment and software assurance classification assessment.

e. Report the results to the acquirer SA manager for approval to ensure agreement on the expected SA level.

2.5 Software Assurance Engineers

2.5.1 Persons performing the function of SA shall work with a project/subproject/program to review, analyze, advise, and report on the software development process.

2.5.2 Persons performing the function of SA shall report all mission-critical and safety-critical findings to the Office of SMA, as well as project, subproject, and/or program management.

CHAPTER 3. Procedures for Acquirer

3.0 The procedures in this section relate to how support is provided by the SA function. A corresponding flowchart is shown in Appendix C. This section applies to both in-house GRC projects and GRC projects contracting out software.

> *Note: For ease of documentation, in the chart below, project/program/subproject/task managers will be referred to simply as project manager.*

3.1 When SA is determined to be needed based on software classification performed on the software to be developed, the SMA manager shall identify an SA manager.

3.2 During project initiation and preaward phase, the SA manager shall perform the following SA activities:

3.2.1 Ensure completion of the SA classification assessment in NASA-STD-8739.8 (Appendix A) for each project and obtain the program/project/subproject/task management agreement on the results. Any disagreements in software classification will need to be resolved by the assurance and engineering independent technical authorities.

a. Perform an independent software classification of the software. Resolve any disagreement with project through appropriate software and SMA technical authorities.

b. Determine safety criticality of the software using the Safety Litmus Test in the NASA-STD-8739.8 (Appendix A1). Obtain concurrence with project.

c. Ensure Class A or safety-critical software is reported to the IV&V facility and entered into NASA software inventory.

3.2.2 Use Appendix F (of this document) to tailor SA requirements for the project based on results from the SA classification assessment and apply the tailored requirements to the acquirer's and provider's SA activities. Document the tailored SA requirements and activities.

3.2.3 Review Request for Proposal (RFP) or Memorandum of Agreement/Understanding (MOA)/MOU) to:

a. Assure contractual statements include appropriate oversight/insight requirements, including needed deliverables such as records, documents, reports, etc.

b. Verify that software quality metrics are addressed.

3.2.4 Prepare a preliminary acquirer SA plan following the template outline in NASA-STD-8739.8 (Appendix B).

3.2.5 Participate in risk identification, analysis, tracking, and control.

3.3 Post RFP, the SA manager shall perform the following preaward SA activities:

3.3.1 Evaluate proposals to verify that the SA requirements in the RFP have been addressed.

3.3.2 Participate in preaward survey if requested.

3.3.3 Participate in contract negotiation.

3.3.4 Update SA classification assessment with the accepted proposal information, then use the updated SA classification assessment to update SA requirements accordingly.

3.4 Post award, the SA manager shall perform the following SA activities:

3.4.1 Update and baseline the acquirer SA plan to reflect changes in the updated SA classification assessment and requirements.

3.4.2 Review the provider’s SA plan to verify that it meets contractual requirements, is compatible with the acquirer SA plan, and is baselined.

3.4.3 Ensure that the acquirer SA personnel are trained and qualified to accomplish their tasks.

3.4.4 Assure that the provider SA personnel are trained and qualified to accomplish their tasks.

3.5 The project manager shall approve the SA Plan and proceed to step 3.8. The provider shall begin its software development effort at this time. However:

3.5.1 If there are issues with the SA Plan, the project manager shall work out minor concerns directly with the SA manager, and then go back to step 3.4 to make necessary changes.

3.5.2 If there are issues which cannot be resolved at this level, go to steps 3.6 and 3.7 to take the issues up to the management chains of both the assurance and the program/project/subproject management.

3.6 The project manager/SA manager shall resolve SA plan issues.

3.7 The assurance and program/project/subproject management shall work the issues, and then go back to step 3.4 to revise the SA Plan.

3.8 The SA manager shall perform contract monitoring per the provider's development lifecycle.

3.8.1 Verify that the provider has developed and maintained processes for assurance of COTS, modified off-the-shelf (MOTS), and GOTS software, addressing both the basic acquired software and any modifications or applications written to adopt them into the intended system.

3.8.2 Assure that both deliverable and any designated nondeliverable software development products including SA records (as identified in Section 1.2) have proper configuration management.

3.8.3 Perform SA tasks to provide insight/oversight of the provider's management, assurance, and engineering processes. Ensure that the project is ready for the next development life cycle.

3.9 The SA manager shall:

3.9.1 Reevaluate the safety criticality of the software.

3.9.2 Determine if the software is safety-critical. The safety criticality has been previously determined in the SA classification assessment. The SA classification assessment shall be updated to reflect any change during reevaluation of the safety criticality of the software.

3.10 If software is determined to be safety-critical, the SA manager/project manager shall perform required software safety functions to assure that:

3.10.1 The requirements in NASA-STD-8719.13 are implemented, including creation and approval of a software safety plan.

3.10.2 Software safety tasks are coordinated between system safety program, software development, and SA to ensure completion and elimination of duplicate efforts.

3.10.3 With input from safety and reliability, software safety analyses (software failure modes and effects analysis, software fault tree analysis, software hazard analysis, and safety data packages) are performed to determine the safety role the software needs to perform, the extent of any potential hazards, and any additional hazards due to design, implementation process decisions, possible design strategies, etc.

a. System safety is notified of any potential hazards, safety-critical functions, or issues.

b. The result of these software safety analyses, along with the safety requirements in the NASA-STD-8719.13, are incorporated into the software requirements and design process.

3.10.4 During design, safety requirements and control of hazards are built in using appropriate software safety features.

3.10.5 The V&V documents are reviewed to assure coverage of safety controls and features. Each software document is reviewed to both assure no additional hazards are present and to gather information to further analyze the hazard potential and mitigation strategies.

3.10.6 Software safety controls, caution, and warnings are verified. Separate safety-specific test plans and procedures are written and performed or appropriate levels of testing is reviewed and witnessed in order to determine full incorporation of all safety requirements, hazard control, and mitigation strategies.

3.10.7 Periodic reviews/audits are conducted for compliance with the defined software safety process for acquisition, development, and assurance of safety-critical software.

3.11 The SA manager/SA engineer shall perform the following SA functions:

3.11.1 Assure that all of the required plans are documented, adhere to applicable standards and procedures, are mutually consistent, and are being executed.

3.11.2 Assure that system documentation is reviewed. System concepts, requirements, and design documents are reviewed to determine and assess the proper incorporation of system-level requirements into

the software. In addition, system-to-software inconsistencies are revealed and reported to project management to determine where the changes need to be made.

3.11.3 Assure participation in risk analysis and perform any assigned action items that fall to SA as a result of risk mitigation, tracking, and assessment.

3.11.4 Assure that software products and related documentation including software planning documents, requirements, design, code, test plans, and test procedures are reviewed to assure conformity and correctness to applicable policies, processes, procedures, and standards and to verify the correct and complete traceability of software requirements from one lifecycle phase to another. Any inconsistencies, errors, problems, etc. found are reported.

3.11.5 Assure that project documentation, including plans, procedures, requirements, design, verification documentation, reports, schedules, and records (and any changes to them) are reviewed for impact to the quality of the product.

3.11.6 Participate in formal and informal reviews including reviews, peer reviews, inspections, and milestone reviews. Provide SA input on software products and process for informal reviews and report SA findings for audits and formal reviews.

3.11.7 Witness formal and acceptance software tests to assure that software and system-level requirements are adequately verified. For other software tests, the tests may be witnessed, and/or the test reports are reviewed by SA personnel to assure requirement verification and completeness.

3.11.8 Assure that the software development plan specifies the standards and procedures for management, acquisition, engineering, and assurance activities.

3.11.9 Ensure that fault tolerance and redundancy have been specified, implemented correctly, and verified by testing.

3.11.10 Assure that software quality metrics (e.g. defect metrics) and process (including trending) are in place and are used to ensure the quality, reliability, and safety of the software products being developed.

a. Analyze software quality metrics using data collected by the project and SA. Examples of software quality metrics include defect type, location, count, priority/criticality, and removal time.

b. Perform trending analyses on the collected software quality metrics and report results to project management and product assurance manager (PAM), including any software reliability analysis and measurements.

c. Assure that the software quality metrics process is documented, monitored, tracked, and assessed for effectiveness and compliance to appropriate documentation (e.g. plans and procedures) or requirements.

3.11.11 Assure that all software management, engineering, development, and assurance processes are audited based on agreed-upon schedule for compliance with applicable standards and plans. The software development folders are audited for completeness. All problems found are documented, tracked, and resolved through the problem reporting and corrective action process and through discussion with project management.

3.11.12 Assure that software engineering practices, development environment, test environment, and libraries employed for the project adhere to applicable standards and procedures.

3.11.13 Assure that software V&V activities occur according to established plans, policies, procedures, and standards. The SA shall collect and maintain SA records showing the participation of SA staff in software V&V efforts, such as minutes, records, artifacts, and signature on test reports.

3.11.14 Ensure that the functional configuration audit and physical configuration audit are performed.

3.11.15 If the project is selected for IV&V, then

a. Assure an IV&V Project Execution Plan (IPEP) is developed by IV&V facility as stated in NPR 7150.2.

b. Verify IV&V work is performed by the contractors selected and managed by IV&V facility

c. Assure that the software developer provides required data and information to NASA IV&V, including specifying on the contracts IV&V's access to system and software products and personnel.

d. Assure input to SA and feedback to project are provided by IV&V team according to the IPEP.

3.11.16 Assure that problems, discrepancies, test anomalies, and risks are recorded, reported (at formal and informal reviews), addressed, analyzed, and tracked to resolution.

3.11.17 Ensure that any acquirer facilities are prepared to receive and install the software prior to delivery.

3.11.18 Assure that coding methods and/or standards are established and followed.

3.12 The SA manager shall determine if the software products meet requirements and plans. Provide project management with a detailed report and make presentations at formal reviews on the status and quality of the software and the adherence to stated plans and procedures. Provide objective evidence to the project and SMA of the readiness of the software product for the next development lifecycle or for final release. If ready for the next lifecycle, go back to step 3.8. If ready for final release, go to step 3.14.

3.13 After software engineering incorporates changes from reviews, inspections, audits, problems reports, corrective actions, engineering change requests, advice, risk analysis, and other analyses, the SA manager shall assure all changes made are properly integrated, tracked, managed, verified, and validated, and that no new problems arise from corrections. Assure all affected products, including documentation, are consistent and correct.

3.14 The project manager, based on all inputs, shall determine if the software is to be released. If not approved, go back to step 3.13 for rework.

3.15 The SA manager shall assure that acquisition knowledge and lessons learned are recorded and entered into the NASA lessons learned database.

> *Note: Software products shall be released to either internal or external customers. The software may go back to systems engineering and from there begin another iteration.*

CHAPTER 4. Procedures for Provider

4.0 The procedures in this section relate to how support is provided by the SA function. A corresponding flowchart is shown in Appendix D. This section applies to both cases when GRC is provider only or is acquirer and provider at the same time. In the latter case, Chapter 3 also applies.

> *Note: For ease of documentation, in the chart below, project/program/subproject/task managers will be referred to simply as project manager.*

4.1 When SA is determined to be needed, based on acquirer SA classification assessment, the project manager shall assign a SA manager who is responsible for directing and managing the SA program.

4.1.1 The SA manager shall have the approval authority role on the establishment and composition of all software baselines.

4.1.2 When selecting to provide its own SA functions, project shall ensure required SA training requirements are met and use GRC SMA as SA's independent reporting channel.

4.2 The SA manager shall establish the SA program, management, and training.

4.2.1 The SA Program:

a. The SA manager shall plan, document, implement, and maintain a SA program for software development, operation, and maintenance activities, which combines the disciplines of software quality, software safety, software reliability, and software V&V.

b. The SA program shall include documentation of SA procedures, processes, tools, techniques, and the methods to be used to assure the quality and safety of the software being developed.

c. The SA program shall include SA processes for acquiring, modifying, and incorporating COTS, MOTS, and GOTS software.

d. The SA program shall describe SA metrics to be collected for the SA program activities.

4.2.2 The SA management:

a. The SA manager shall establish and maintain interfaces with project management and ensure the working relationship between SA personnel and that of the project.

b. The SA manager shall maintain a reporting channel to the GRC Quality Engineering and Assurance Branch and PAMs, which are independent of project management and software development.

4.2.3 The SA training:

a. The SA manager shall ensure that personnel managing, developing, and implementing the SA process are trained and/or experienced in SA. Ensure that training records are maintained.

b. The SA training, including relevant software engineering design methods and languages, processes, development environments, tools, test techniques, and other software engineering and assurance methods, shall be obtained and/or originated and maintained for management, engineering, and assurance personnel for the engineering environment and products they assure.

c. Training shall be provided for the environment and operational particulars of the programs/projects to which they are assigned.

4.3 The SA manager shall write and maintain a SA plan.

4.3.1 The SA Plan addresses all software development and maintenance activities. Any proposed changes to the baselined SA Plan shall be submitted to the acquirer as a formal change request accompanied by a risk analysis (per NPR 7120.5) conducted to identify the potential impact of the change.

4.3.2 This SA Plan shall conform to IEEE 730-2002, IEEE Standard for Software Quality Assurance Plans.

4.3.3 In addition, this SA Plan shall address how the provider implements the requirements of Section 6 and 7 of the NASA-STD-8739.8, as well as requirements in this document including records.

4.3.4 In the event there are conflicts between implementation of the requirements of Section 6 and 7 of the NASA-STD-8739.8 and the IEEE 730-2002, the NASA-STD-8739.8, shall take precedence.

4.4 The project manager shall approve the SA Plan and:

4.4.1 Proceed to step 4.7.

4.4.2 The software development effort shall begin at this time, using the GLPR 7150.1, Software Engineering Requirements.

4.5 The project manager shall work out minor concerns with the SA Plan directly with the SA manager and the provider then go back to step 4.3 to make the necessary changes. If there are issues which cannot be resolved at this level, go to step 4.6 to escalate the issue(s) to the GRC Quality Engineering and Assurance Branch and program/project/subproject management.

4.6 The GRC Quality Engineering and Assurance Branch and the program/project/subproject management shall work the SA plan issues then go back to step 4.3 to revise the SA Plan.

4.7 The SA manager/SA engineers shall provide SA to development processes.

4.8 The SA manager shall determine if the software is safety-critical per the SA classification assessment.

4.9 The SA classification assessment shall be updated to reevaluate the safety criticality of the software.

4.10 If software is determined to be safety-critical, the SA manager/project manager shall perform the required software safety functions to assure that:

a. The requirements for NASA-STD-8719.13 are implemented, including creation and approval of a software safety plan.

b. Software safety tasks are coordinated between system safety program, software development, and SA to ensure completion and elimination of duplicate efforts.

c. Software safety analyses (software failure modes and effects analysis, software fault tree analysis, software hazard analysis, and safety data packages) are performed, with input from safety and reliability, to determine the safety role the software needs to perform, the extent of any potential hazards, any additional hazards due to design and implementation process decisions, possible design strategies, etc.

(1) System safety is notified of any potential hazards, safety-critical functions, or issues.

(2) The results of these software safety analyses, along with the safety requirements in the NASA-STD-8719.13, are incorporated into the software requirements and design process.

4.10.1 During design, safety requirements and control of hazards are built in using appropriate software safety features.

4.10.2 The V&V documents are reviewed to assure coverage of safety controls and features. Each software document is reviewed to ensure no additional hazards are present, to gather information, and to further analyze the hazard potential and mitigation strategies.

4.10.3 Software safety controls, cautions, and warnings are verified. Write/perform separate safety-specific test plans and procedures, or review and witness appropriate levels of testing in order to determine full incorporation of all safety requirements, hazard control, and mitigation strategies.

4.10.4 Periodic reviews/audits are conducted for compliance with the defined software safety process for acquisition, development, and assurance of safety-critical software.

4.11 The SA manager/SA engineer shall perform the following SA functions:

4.11.1 Assure that all of the required plans are documented, adhere to applicable standards and procedures, are mutually consistent, and are being executed.

4.11.2 Assure that system documentation, system concepts, requirements, and design documents are reviewed to determine and assess the proper incorporation of system-level requirements into the software. In addition, system-to-software inconsistencies are revealed and reported to project management to determine where the changes need to be made.

4.11.3 Assure participation in risk analysis and perform any assigned action items that fall to SA as a result of risk mitigation, tracking, and assessment.

4.11.4 Assure that software products and related documentation, including software planning documents, requirements, design, code, test plans, and test procedures, are reviewed to ensure conformity and correctness to applicable policies, processes, procedures, and standards and to verify the correct and complete traceability of software requirements from one lifecycle phase to another. Any inconsistencies (errors, problems, etc.) found are then reported.

4.11.5 Assure that project documentation, including plans, procedures, requirements, design, verification documentation, reports, schedules, and records (and any changes to them) are reviewed for impact to the quality of the product.

4.11.6 Participate in formal and informal reviews, including peer reviews, inspections (e.g. serving as the moderator), and milestone reviews. The SA provides input on software products and processes for informal reviews. The SA reports findings from audits and formal reviews.

4.11.7 Witness formal and acceptance of software tests to ensure that software and system-level requirements are adequately verified. For other software tests, the tests may be witnessed and/or test reports reviewed by SA personnel to ensure requirement verification and completeness.

4.11.8 Assure that the software development plan specifies the standards and procedures for management, acquisition, engineering, and assurance activities.

4.11.9 Ensure that fault tolerance and redundancy have been specified, implemented correctly, and verified by testing.

4.11.10 Assure that software quality metrics (e.g. defect metrics) and process (including trending) are in place and are used to ensure the quality, reliability, and safety of the software products being developed.

a. Analyze software quality metrics using data collected by the project and SA. Examples of software quality metrics include defect type, location, count, priority/criticality, and removal time.

b. Perform trending analyses on the collected software quality metrics and report results to project management and PAM, including any software reliability analysis and measurements.

c. Assure that the software quality metrics process is documented, monitored, tracked, and assessed for effectiveness and compliance to appropriate documentation (e.g. plans and procedures) or requirements.

4.11.11 Assure that all software management, engineering, development, and assurance processes are audited, based on agreed upon schedule, for compliance with applicable standards and plans. Software development folders are audited for completeness and all problems found are documented, tracked, and resolved through the problem reporting and corrective action process and through discussion with project management.

4.11.12 Assure that software engineering practices, development environment, test environment, and libraries employed for the project adhere to applicable standards and procedures.

4.11.13 Assure that software V&V activities occur according to established plans, policies, procedures, and standards. The SA collects and maintains the SA records showing the participation of SA staff in software V&V efforts, such as minutes, records, artifacts, and signatures on test reports.

4.11.14 Ensure that functional configuration audit and physical configuration audits are performed to ensure that all deliverables are present and in proper shape for release and that all requirements have been met.

4.11.15 If the project is selected for IV&V, then:

a. Assure an IPEP is developed by IV&V facility, as stated in NPR 7150.2.

b. Verify IV&V work is performed by the contractors selected and managed by IV&V facility.

c. Assure that required data and information is provided to NASA IV&V.

d. Assure input to SA and feedback to project are provided by IV&V team according to the IPEP.

4.11.16 Assure that problems and risks are recorded, reported (at formal and informal reviews), addressed, and tracked to closure.

4.11.17 Ensure that SA records (as specified in Section 1.2) are collected, maintained, and placed under configuration management.

4.11.18 Assure that coding methods and/or standards are established and followed.

4.12 The SA manager shall determine if the software products meet requirements and follow plans. Provide project management with a detailed report and make presentations at formal reviews on the status, quality of the software, and the adherence to stated plans and procedures. Provide objective evidence to the project and SMA of the readiness of the software product for the next development lifecycle or for final release. If the software is ready for next lifecycle, go back to step 4.7. If ready for final release, proceed to step 4.14.

4.13 The SA manager, after software engineering incorporates changes from reviews, inspections, audits, problems reports, corrective actions, engineering change requests, advice, risk analysis, and other analyses, shall assure all changes made are properly integrated, tracked, managed, verified, and validated, and that no new problems arise from corrections. Ensure all affected products, including documentation, are consistent and correct. Go back to step 4.8.

4.14 Project manager, based on all inputs, shall approve the software for release. If not approved, go back to step 4.13 for rework.

4.15 The SA manager shall assure that acquisition knowledge and lessons learned are recorded and entered into the NASA lessons learned database.

> *Note: Software products shall be released to either internal or external customers. The software may go back to systems engineering and from there begin another iteration.*

Appendix A. Definitions

Note: Many of the terms used within this procedure for software assurance are the same as those used for software development or product assurance and will not be repeated here.

A.1 Acquirer. The entity or individual who specifies the requirement and accepts the resulting software products. The acquirer is usually NASA or an organization within the Agency but can also refer to the prime contractor/subcontractor relationship as well.

A.2 Functional Configuration Audit (FCA). An audit conducted to verify that the development of a configuration item has been completed satisfactorily, that the item has achieved the performance and functional characteristics specified in the functional or allocated configuration identification, and that its operational and support documents are complete and satisfactory.

A.3 Independent Verification and Validation (IV&V). Verification and validation performed by an organization that is technically, managerially, and financially independent. The IV&V, as a part of software assurance, play a role in the overall NASA software risk mitigation strategy applied throughout the lifecycle, to improve the safety and quality of software.

A.4 Insight. Surveillance mode requiring the monitoring of acquirer-identified metrics and contracted milestones. Insight is a continuum that can range from low intensity, such as reviewing quarterly reports, to high intensity, such as performing surveys and reviews.

A.5 Mission-Critical. Item or function that must retain its operational capability to assure mission success

A.6 Oversight. Surveillance mode that is in line with the supplier processes. The acquirer retains and exercises the right to concur or nonconcur with the supplier decisions. Nonconcurrence must be resolved before the supplier can proceed. Oversight is a continuum that can range from low intensity, such as acquirer concurrence in reviews (e.g., Preliminary Design Review (PDR), Critical Design Review (CDR)), to high intensity oversight, in which the customer has day-to-day involvement in the supplier's decisionmaking process (e.g., software inspections).

A.7 Physical Configuration Audit (PCA). An audit conducted to verify that one or more configuration items, as built, conform to the technical documentation that defines it (based on IEEE 610.12, IEEE Standard Glossary of Software Engineering Terminology).

A.8 Process Assurance. Activities to assure that all processes involved with the project adhere to plans and comply with the contract and/or any memorandum of agreement/understanding.

A.9 Product Assurance. Activities to assure that all required plans are documented, and that the plans, software products, and related documentation adhere to plans and comply with the contract and/or any memorandum of agreement/understanding.

A.10 Provider. The entities or individuals that design, develop, implement, test, operate, and maintain the software products. A provider may be a contractor, a university, a separate organization within NASA, or within the same organization as the acquirer. The term “provider” is equivalent to “supplier” in the

International Organization for Standardization/ International Electrotechnical Commission (ISO/IEC) 12207, software lifecycle processes.

A.11 Review. (v.) Read through material; evaluate as to content, clarity, correctness, consistency, compliance, completeness, functionality, performance, level of detail, safety, reliability, traceability, and more, as required; provide input on the product under review; and report on findings and observations verbally, in writing, using review forms, inspection reports, etc. (n.) Reviews such as major milestone reviews and preship reviews at either the system or software level. Can also be lower-level review boards or failure reviews, etc.

A.12 Safety-Critical Software. Software is safety-critical if it meets at least one of the following criteria:

a. Resides in a safety-critical system (as determined by a hazard analysis) AND at least one of the following:

(1) Causes or contributes to a hazard.

(2) Provides control or mitigation for hazards.

(3) Controls safety-critical functions.

(4) Processes safety-critical commands or data.

(5) Detects and reports, or takes corrective action, if the system reaches a specific hazardous state.

(6) Mitigates damage if a hazard occurs.

(7) Resides on the same system (processor) as safety-critical software.

b. Processes data or analyzes trends that lead directly to safety decisions (e.g., determining when to turn power off to a wind tunnel to prevent system destruction).

c. Provides full or partial verification or validation of safety-critical systems, including hardware or software subsystems.

A.13 Software Assurance (SA). A planned and systematic pattern of all actions necessary to provide adequate confidence that item or product conforms to established technical requirements. For NASA, this includes the discipline of software quality (functions of software quality engineering, software quality assurance, and software quality control), software safety, software reliability, software verification and validation, and IV&V.

A.14 Software Development Cycle. The period of time that begins with the decision to develop a software product and ends when the software is delivered (based on IEEE 610.12). It can include all, or combinations of, the following phases: requirements, design, implementation, integration, testing, release (or delivery).

A.15 Software Lifecycle. The period of time that begins when a software product is conceived and ends when the software is no longer available for use. Can include all, or combinations of, the following phases:

concept/exploration, requirements, design, implementation, integration, testing, release (or delivery), operations and maintenance, retirement/termination.

A.16 Software Product. The complete set of computer programs, procedures, and associated documentation and data designated for delivery to a user. The software product includes programs and operational data contained in hardware (e.g., firmware, programmable logic, and programmable gate arrays).

A.17 Software Safety. The discipline of SA that is a systematic approach to identifying, analyzing, tracking, mitigating, and controlling software hazards and hazardous functions (data and commands) to ensure safe operation within a system.

A.18 Software Reliability. The discipline of SA that, (1) defines the requirements for software controlled system fault/failure detection, isolation, and recovery, (2) reviews the software development processes and products for software error prevention and/or reduced functionality states, and (3) defines the process for measuring and analyzing defects and defines/derives the reliability and maintainability factors.

A.19 Testing. An activity in which a system or component is executed under specified conditions, the results are observed and/or recorded, and an evaluation is made of some aspect of the system or component [based on IEEE 610.12].

A.20 Test Plan. (1) (IEEE Std 829-1983 [5]) A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do each task and any risks requiring contingency planning, and (2) a document that describes the technical and management approach to be followed for testing a system or components (based on IEEE 610.12).

A.21 Test Procedure. (1) Detailed instructions for the setup, execution, and evaluation of results for a given test case, and (2) A document containing a set of associated instructions as in (1) (based on IEEE 610.12).

A.22 Validation. Validation provides objective evidence that the product meets the intended use of the product. Validation follows successful verification activities and may include system readiness reviews, test readiness reviews, operational readiness reviews, inspections, and testing. The process of evaluating a system or component during, or at the end of, the development process to determine whether or not it satisfies the customer (i.e., ultimately, was the right product built). A customer-approved design review and/or acceptance test can serve as product validation.

A.23 Verification. Verification provides evidence that the design or system meets the input requirements. This evidence may consist of alternative calculations showing similarity with a proven design, peer, or outside design reviews, analytical simulations, and test results.

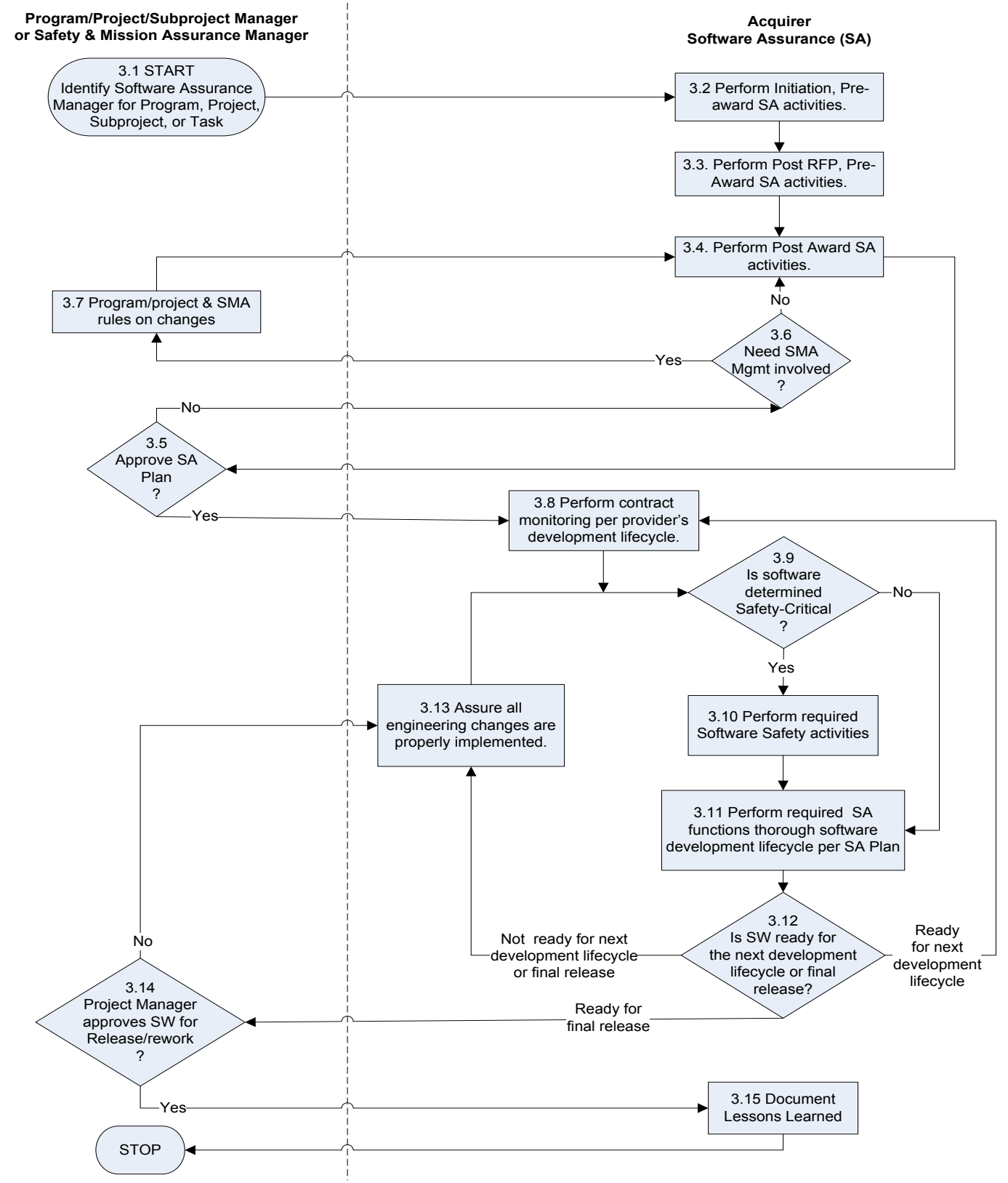
a. The process of evaluating a system or a component to determine whether the products of a given development phase satisfies the conditions imposed at the start of that phase.

b. Formal proof of program correctness (i.e., were the correct processes, standards and procedures followed and followed correctly, was it done the right way and all the prescribed products produced in the manner and extent required, is it being built correctly, does the design document reflect all the requirements, does final software build have all the required functionality and performance).

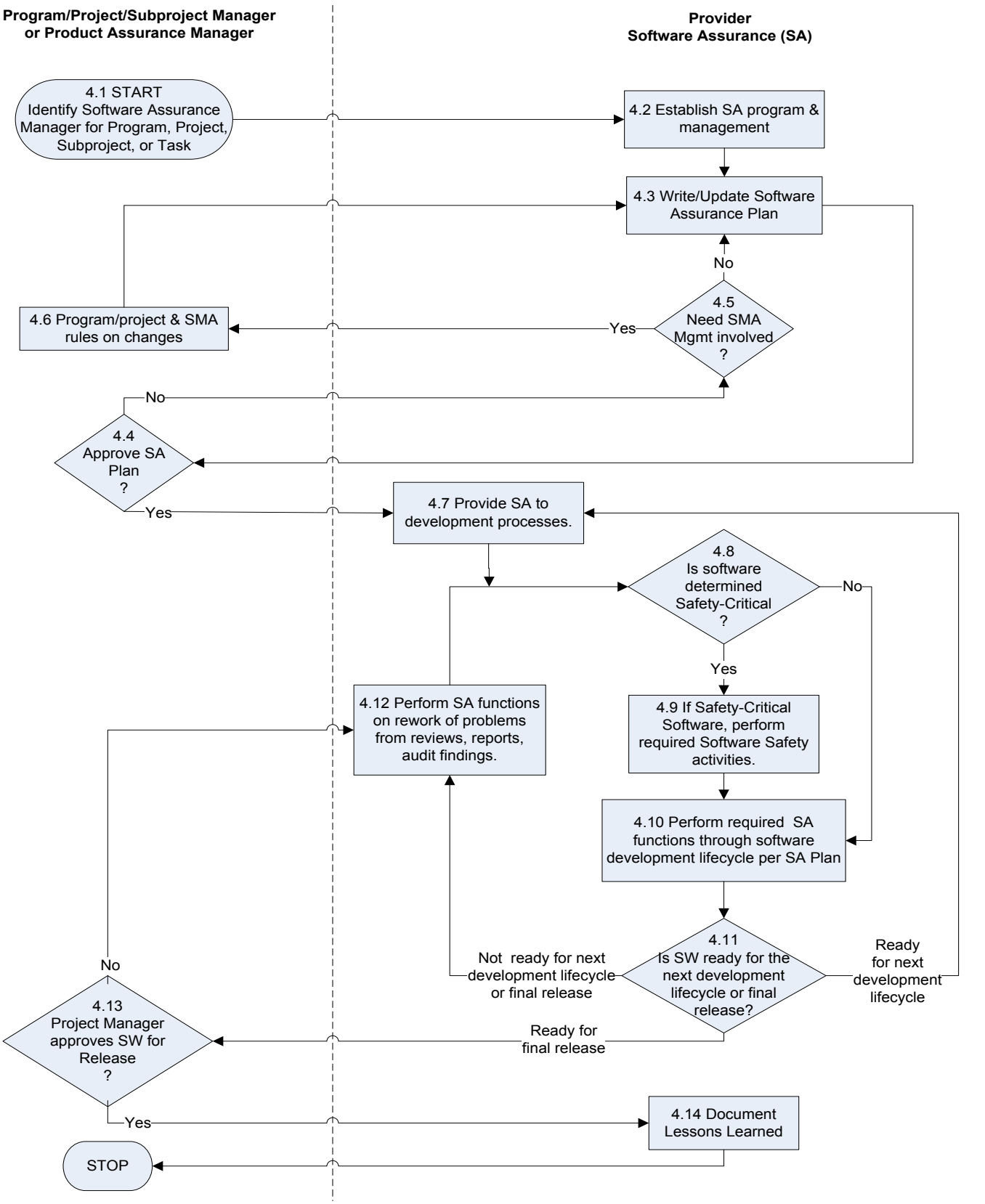
Appendix B. Acronyms

B.1	COTS	Commercial-off-the-shelf
B.2	GLPR	Glenn Procedural Requirements
B.3	GOTS	Government-off-the-shelf
B.4	GRC	Glenn Research Center
B.5	IEEE	Institute of Electrical and Electronics Engineers
B.6	IPEP	IV&V Project Execution Plan
B.7	ISO/IEC	International Organization for Standardization/ International Electrotechnical Commision
B.8	IV&V	Independent Verification and Validation
B.9	MOA	Memorandum of Agreement
B.10	MOTS	Modified off-the-shelf
B.11	MOU	Memorandum of Understanding
B.12	NPD	NASA Policy Directive
B.13	NPR	NASA Procedural Requirements
B.14	PAM	Product Assurance Manager
B.15	PAP	Product Assurance Plan
B.16	REDAA	Requirement Engineering Design Audits and Assessments
B.17	RFP	Request for Proposal
B.18	SA	Software Assurance
B.19	SACA	Software Assurance Classification Assessments
B.20	SAP	Software Assurance Plan
B.21	SMA	Safety and Mission Assurance
B.22	SMAD	Safety and Mission Assurance Directorate
B.23	STD	Standard
B.24	QAAR	Quality Audit, Assessment, and Reviews
B.25	V&V	Verification and Validation

Appendix C. Flowchart for Acquirer



Appendix D. Flowchart for Provider



Appendix E. Guidance

Note: This section is meant only to provide guidance and insight into some of the software assurance activities and documentation required in Section 1.2 of this document. It is for guidance only.

E.1 Analysis Results:

The software assurance (SA) analyses could be problem or error trending, criticality analyses to determine if and what critical functions the software may be responsible for and to what level, such as Software Fault Tree Analyses, Software Formal Inspection metrics comparisons or trends, Software Failure Modes and Effects Analyses, etc.

E.2 Audits:

a. For low SA efforts, no audits are usually performed unless specifically negotiated and recorded in the Software Management Plan (SMP) and/or Software Assurance Plan (SAP). For medium SA efforts, only a Functional Configuration Audit (FCA) and a Physical Configuration Audit (PCA) are performed. Both are performed during acceptance, and prior to, release of the software. The FCA often utilizes the Acceptance Test results and possibly some lower level testing to assure all requirements are met. The PCA is verification that all the physical deliverables are ready, and in a known state prior to release.

b. For full SA efforts, besides the FCA and PCA, additional audits (such as process audits for processes of the areas specified in Capability Maturity Model Integrated (CMMI) Maturity Level 2 or 3) should be performed during the design phase of the software lifecycle. These additional audits focus on determining if the program/project/subproject/task is following its plan, procedures, standards, and guidelines. The number of audits and their focus is determined with program/project/subproject/task management and written into the SAP. For instance, there might be an audit of just the configuration management process for a program/project/subproject/task or an audit might cover usage of software standards and guidelines. For most software efforts, an audit of this type tries to cover all the major procedures to some degree. If the program/project/subproject/task is a large project, distributed over many groups and/or companies or perhaps spans more than 4 years in duration, then additional or more focused audits should be considered and described in the SA Plan.

E.3 Audit/Assessment Results:

a. Audit documentation can be as simple as a report generated summing up the audit coverage and the auditor findings and observations. This would be in the case of low or medium SA intermediate audits.

b. At the other end of the spectrum, there can be audit checklists, both program/project/subproject/task specific and/or lifecycle phase or subject matter specific, reports, records, and closeout reports. There can be forms created from these checklists and the filled out forms then constitute part of the audit record. An audit report then summarizes what was audited, the findings and observations, and expected time period for findings to be addressed, and the program/project/subproject/task is ready for a follow-up audit. The closed-out audit report shows the date and signatures of those involved as well as showing closure for all findings. The FCAs and PCAs require this level of rigor.

E.4 Discrepancy Reports:

These are reports of problems found with the software process, equipment, test setup, etc. Depending on the project, equipment, and test/setup types of discrepancies may be written as problem reports especially if on final test hardware or acceptance testing. In general, these are reports the SA engineer would make when detecting problems with the software process, or when the program/project/subproject/task is not following their procedures and plans.

E.5 Inspection Reports:

When formal inspections are performed on a software product, there are several records that are generated: Inspection Announcement Record, Individual Preparation Logs, Inspection Defect List, Inspection Summary Report, and Detailed Inspection Report. These will normally be kept by the software program/project/subproject/task lead, but SA may be asked (in the PAP or informally) to keep these records for a program/project/subproject/task.

E.6 Document Review Inputs:

a. This refers to the reports and records created to show proof of SA involvement in the software and system review process. When it is an official review of a baselined document for which SA has signoff concurrence or approval, the signature on that document may be the only record needed. However, it may also cover the records for review of pre-baseline product documentation such as requirements, specifications, design, management plans, test plans, test procedures, etc.

b. The SAP and/or a division, program, project or organization's SA procedures will indicate which formal records are needed, where they are kept, on what documents they will be officially used, and what forms the records may take, paper or electronic copies of redlined documents, document review record forms, meeting minutes, etc.

c. It is recommended that SA review and concur on software management planning documents, especially configuration management, risk management, verification, and validation plans. The SA also needs to review and provide input on software requirements and design documentation as well as test procedures and certification procedures. While SA usually has no official signoff for most of these documents, it is necessary to review them and SA is often provided draft copies so that their inputs can be incorporated. It is necessary however, to keep records of these inputs especially if they are not incorporated for some reason. These review records, formal or informal, provide proof of SA coverage of a software effort and provide a record and insight into program/project/ subproject/task management and SA concurrence or nonconcurrence.

E.7 Software Quality Assurance Procedures:

This document could consist of a division, program, project, or an organization procedure for performing quality assurance activities. It may include SA processes, records, checklists, and policies beyond what is required and written in a specific SA plan. It might include things like procedures and guidelines for how and when to conduct audits, perform document reviews, trending analysis, etc. Specific assessments for risk management, verification and validation, problem reporting, corrective action, Data Acceptance Packages, Version Description Documents, ISO compliance, etc., can be a part of SA procedures.

E.8 Inputs to Reviews and Boards:

a. The SA is often required or requested to present their findings and observations at a formal review, e.g. Requirements Definition Review (RDR), Preliminary Design Review (PDR), Critical Design Review (CDR), Test Readiness Review (TRR), Reflight Review, Acceptance Review, Pre-Ship Review (PSR), etc. When called upon to provide input for these reviews, the presentation itself is the record, and a copy should be kept in the SA files for that program/project/subproject/task. In addition, a copy, or pointer to the official copy location, of any minutes for these milestone reviews should be kept. If called upon to be a board member, the SA engineer should keep a record of the request to serve as a board member for that software or system effort. Any other records will be part of the program/project/subproject/task review records.

b. For full SA projects especially, SA is often a member of the change control board. As a member of these boards, board minutes will serve as records of SA involvement and these records are kept by the program/project/subproject/task.

E.9 Software Assurance Plan (SAP):

a. The purpose of creating a SAP is to document and specify the conduct of the activities that will comprise SA for a specific project. This plan establishes procedures for performance of SA activities on each individual project. The SAP describes in detail the specific activities to be performed by project management, software engineering, and software assurance engineering in order to assure that a given software system meets its quality requirements. The plan serves as a contract or agreement between project management and the SMA Directorate (SMAD), or other SA provider, concerning the software management and assurance activities.

b. This plan is developed in accordance with GLPR 7150.1 GRC Software Engineering Requirements. As a minimum, the SAP should contain the process activities for reviews of documentation and source code, performance verification, criteria for the completion of specific development stages, configuration management assurance, and maintenance of records. The SAP may be a "standalone" document or may be a section of an overall Product Assurance Plan (PAP). In either case, the SAP is considered a part of the overall PAP.

E.10 Applicable Development Standards:

The project manager or the software lead chooses standards for the software development process. All applicable standards, once chosen and agreed to, (i.e., language, documentation, object oriented development tools, etc.) should be adhered to throughout the entire software development lifecycle. Some applicable standards can be found in the GLPR 7150.1 GRC Software Engineering Requirements. If necessary, a modification to these standards can be obtained by written agreement between the project and the SMAD or the SA provider via the PAP or SAP.

E.11 Use of Government, Commercial, Contractors or Subcontractors Software:

a. Any software provided as government-furnished equipment or Government Off-the-Shelf (GOTS) shall meet project requirements to the extent that it was intended. In part or in whole, any software obtained from contractors or subcontractors shall be developed under a contract which specifies that they follow GRC procedures or higher.

b. Commercially purchased software (COTS) shall be configuration managed and tested to the extent possible. All documentation (including the licenses), the executable and any libraries, support software, etc. for COTS software should be kept under configuration management.

c. The following items from the provider/vendor can be examined to determine if the GOTS or COTS has received adequate software quality assurance support:

	Item to look at	What to look for
Corporate Level	Corporate Quality Manual	Software quality organization, formal quality program certification (ISO/CMMI).
	Software Quality Assurance Program	Software assurance procedures, work instructions, personnel, resources.
	Configuration Management (CM) System	How software assurance support the build/baseline processes? What are the frequency of CM audits and the overall effectiveness?

Product Level	User Manual	Quality statement, application domain, compatibility, integration, interoperation, security, library resources, support service.
	Certification	Quality stamp/signature, quality statement, authentication.
	Warranties	Limitation, warranty period, upgrades,
	V&V Documents	Test results under the operational profile of past user, track records. Test procedures and reports.

E.12 Firmware Creation and Installation:

The SA should apply to the development and support of firmware source code and its documentation. The development and assurance of firmware is considered no different from the development and assurance of software. A process shall be developed to include procedures for properly burn-in/install/uninstall of source code and verify the correct version.

Appendix F. Tailoring of Requirements

Table 1 of NASA-STD-8739.8: Example of Tailoring for Software Assurance Requirements

Class	A	B	C	D	E	F,G,H
Effort	Full	Full	Medium	Minimal	N/A at this time	Not Covered
Tailoring of Requirements	All software assurance requirements apply with no tailoring	All software assurance requirements apply – some minor tailoring to meet project objectives & mission category	Medium tailoring of software assurance requirements to meet project objectives & mission category	Major tailoring of software assurance requirements to meet project objectives & mission category	Initial Classification survey periodically to assure project remains a Class E software project	N/A unless requested
To what extent Requirements are Met*	All activities to meet these requirements will be performed.	All activities to meet the requirements will be performed, how to meet the requirements may be less rigorous.	Activities to meet the requirements may be tailored, i.e., how to meet the requirements will be less rigorous.	Activities to meet the requirements will be tailored, i.e., how to meet the requirements will be minimal.	Activities will mostly consist of assurance reports on project classification unless otherwise contracted/agreed	Only as specified in an agreement
*How the requirements will be implemented, level of rigor to which the requirements are met.						

F.1 This table is provided to be used as a guide for tailoring Software Assurance (SA) requirements based on the results of the SA Classification Assessment. There are two parts of this table. The first part is to determine the applicable SA requirement areas. Then, the second part is to determine the specific requirements for that area and the level of rigor to which the requirements are implemented.

Appendix G. Compliance Matrix

G.1 The compliance matrix table from NASA-STD-8739.8 is included to aid this analysis. Start with the requirements section of the table to determine if applicable. Then, go to the requirement column of that section to assess which requirements are needed and what level of rigor should they be implemented based on the Software Assurance Classification Assessments (SACA) results, the tailoring table, and the project mission objectives. The block column shows the corresponding block number in this document (Chapter 3 and 4) to the requirement in NASA-STD-8739.8 and is used to identify application of Software Assurance (SA) requirements to SA activities. The end products of this process are a set of tailored SA requirements and corresponding SA activities ready to be placed in the SA Plan or similar document.

a. Example:

1. Post Request for Proposal (RFP), preaward SA requirements area is needed.
    - Req. 5.2.1.1 Evaluate proposal (needed-tailored in).
        a. Implementation: Technical area expert with nonvoting member.
    - Req. 5.2.1.2 Participate in preaward surveys when such surveys are requested (no need-tailored out).
    - Req. 5.2.1.3 Participate in contract negotiation (no need-tailored out).
    - Req. 5.2.1.4 Perform an updated SACA (needed–tailored in).
        a. Implementation: Update the SACA when there is a major change in software, system, mission objective, and at every project milestone reviews.
    - Req. 5.2.1.5 Update SA requirements based on assessment results (needed–tailored in).
        a. Implementation: The SA requirements shall be updated within 7 business days after the SACA results are available.
    - Req. 5.2.1.6 Maintain Assessment results (needed–tailored in)
        a. Implementation: The SA manager shall maintain all SACA results in accordance with office's quality record procedure.

    Note 1: For Class B software the actual requirements are not tailored but the implementation can be tailored to some degree. Class C software may address tailoring the assurance requirements based on what is applicable for the software engineering requirements of NPR 7150.2 and according to any potential risks specific to the planned operational or development environment. Class D software may have the most requirements tailoring, matching the assurance activities to the less formal development activities. The SA manager must work closely with the project to assess the software for the project and tailor the SA activities accordingly.

    Note 2: Often Class D assurance activities consist mostly of assuring any contractual agreements meet the needs of the project/program and then performing periodic audits and surveys of the project work to follow up. The level of SA effort applied to any class is commensurate with the risk, criticality, complexity, and needed reliability and quality of a project.

    Note 3: If the results of the SACA identify the software as Class E (which excludes the exploratory software), then the requirements of this procedure are not mandatory.

    Note 4: Class F, G and H software are currently the responsibility of the Chief Information Office, however, for the higher level of Information Technology or business systems class, if SA is requested, those projects would be assured in accordance with the software engineering requirements in NPR 7150.2.

NASA-STD-8739.8 Requirements					
				GLPR 8739.1	
Section	No.	Requirement	Role/Resp	(C)	Block
Acquirer Software Assurance	5	*Not a requirement*			
Initialization, Preaward	5.1	*Not a requirement*			
	5.1.1	Identify software assurance manager	Acquirer SMA Mgr	F	3.1
	5.1.2	The software assurance manager shall ensure the following tasks:			
	5.1.2.1	Complete Software Classification Assessment	Acquirer SA Mgr	F	3.2.1
	5.1.2.2	Safety-critical software compliance from assessment	Acquirer SA Mgr	F	2.3, 3.10.1
	5.1.2.3	Complete coverage of sections 5, 6, & 7.	Acquirer SA Mgr	F	3.2.2
	5.1.2.4	Project agreement with classification	Acquirer SA Mgr	F	3.2.1
	5.1.2.5	Application of software assurance requirements for acquirer	Acquirer SA Mgr	F	3.2.2
	5.1.2.6	Application of software assurance requirements for provider of each MOU/MOA	Acquirer SA Mgr	F	3.2.2
	5.1.2.7	Contractual statements include insight/oversight	Acquirer SA Mgr	F	3.2.3
	5.1.2.8	Prepare preliminary acquirer program/project software assurance plan	Acquirer SA Mgr	F	3.2.4
	5.1.2.9	Verify that the RFP/MOU/MOA addresses software quality metrics	Acquirer SA Mgr	F	3.2.3
	5.1.2.10	Identify, analyze, track, and control procurement/development risks	Acquirer SA Mgr	F	3.2.5
Post RFP, Pre-Award	5.2	Not a requirement			
	5.2.1	The software assurance manager shall perform the following tasks:			
	5.2.1.1	Evaluate proposals	Acquirer SA Mgr	F	3.3.1

	5.2.1.2	Participate in preaward surveys when such surveys are requested	Acquirer SA Mgr	F	3.3.2
	5.2.1.3	Participate in contract negotiations	Acquirer SA Mgr	F	3.3.3
	5.2.1.4	Perform an updated Software Assurance Classification Assessment	Acquirer SA Mgr	F	3.3.4 3.9
	5.2.1.5	Update software assurance requirements based on Assessment results	Acquirer SA Mgr	F	3.3.4
	5.2.1.6	Maintain Assessment results	Acquirer SA Mgr	F	1.2
Post-Award, Pre-Development	5.3	Not a requirement			
	5.3.1	The software assurance manager shall perform the following tasks:			
	5.3.1.1	Verify provider's software assurance plan meets contractual requirements.	Acquirer SA Mgr	F	3.4.2
	5.3.1.2	Verify acquirer's and provider's software assurance plans are consistent, compatible, and are baselined	Acquirer SA Mgr	F	3.4.1
	5.3.1.3	Ensure acquirer software assurance personnel are trained and qualified	Acquirer SA Mgr	F	3.4.3
	5.3.1.4	Assure provider software assurance personnel are trained and qualified	Acquirer SA Mgr	F	3.4.4
Contract Implementation, Development	5.4	Not a requirement			
	5.4.1	The software assurance manager shall perform the following tasks:			
	5.4.1.1	Assure both acquirer and provider software assurance organizations perform according to their plans	Acquirer SA Mgr	F	2.3
	5.4.1.2	Verify provider has developed and maintained processes for assurance of COTS, MOTS, and GOTS software	Acquirer SA Mgr	F	3.8.1
	5.4.1.3	Ensure insight performed over provider	Acquirer SA Mgr	F	2.3 3.8.3
	5.4.1.4	Ensure oversight performed over provider	Acquirer SA Mgr	F	2.3 3.8.3
	5.4.1.5	Assure proper software configuration management	Acquirer SA Mgr	F	3.8.2

	5.4.1.6	Assure software issues are documented and tracked to resolution	Acquirer SA Mgr	F	3.11.16
	5.4.1.7	Assure software products are reviewed and assure that software quality metrics are collected and analyzed.	Acquirer SA Mgr	F	3.11.4, 3.11.10
Acceptance	5.5	*Not a requirement*			
	5.5.1	The software assurance manager shall perform the following tasks:			
	5.5.1.1	Ensure an acceptance audit is performed prior to delivery	Acquirer SA Mgr	F	3.11.14
	5.5.1.2	Ensure that any acquirer facilities are prepared to receive and install the software	Acquirer SA Mgr	F	3.11.17
	5.5.1.3	Assure all acceptance documentation is complete	Acquirer SA Mgr	F	3.11.14
	5.5.1.4	Assure acquisition lessons learned are recorded and entered into the NASA lessons learned database	Acquirer SA Mgr	F	3.15
Operation	5.6	*Not a requirement*			
	5.6.1	The software assurance manager shall perform the following tasks:			
	5.6.1.1	Ensure software assurance processes are in place for operation of the software developed or acquired by NASA	Acquirer SA Mgr	F	2.3
	5.6.1.2	Ensure software assurance processes include a periodic audit of the operational software	Acquirer SA Mgr	F	2.3
	5.6.2	Ensure software configuration management of operational software	Acquirer SA Mgr	F	2.3
Maintenance	5.7	*Not a requirement*			
	5.7.1	The software assurance manager shall perform the following tasks:			
	5.7.1.1	Ensure software assurance processes are in place for software maintenance	Acquirer SA Mgr	F	2.3
	5.7.1.2	Assure transfer and maintenance of any licenses, simulators, models, and test suites	Acquirer SA Mgr	F	2.3
	5.7.1.3	Assure that any software metrics are transferred to the maintenance organization and maintained	Acquirer SA Mgr	F	2.3

Retirement	5.8	*Not a requirement*			
	5.8.1	The software assurance manager shall perform the following tasks:			
	5.8.1.1	Assure that software engineering and management prepare, approve, and execute a retirement plan	Acquirer SA Mgr	F	2.3
	5.8.1.2	Ensure that the retirement plan includes archival or disposal of software assurance records and documents	Acquirer SA Mgr	F	2.3
Provider Software Assurance	6	*Not a requirement*			
Software Assurance Program	6.1	*Not a requirement*			
	6.1.1	Plan, document, and implement software assurance program	Provider SA Mgr	F	4.2.1
	6.1.2	Include software assurance processes for COTS, MOTS, and GOTS software	Provider SA Mgr	F	4.2.1
	6.1.3	Include all software assurance disciplines	Provider SA Mgr	F	4.2.1
	6.1.4	Coordinate with IV&V	Provider SA Mgr	F	4.10.15
	6.1.5	Describe SA metrics collection and reporting	Provider SA Mgr	F	4.2.1
Software Assurance Management	6.2	*Not a requirement*			
	6.2.1	Identify provider software assurance manager	Provider Mgmt	F	4.1
	6.2.2	Establish and maintain interface between software assurance and project	Provider Mgmt, Provider SA Mgr	F	4.2.2
	6.2.3	Establish an independent reporting channel to provider management	Provider Mgmt, Provider SA Mgr	F	4.2.2
	6.2.4	Conduct and document periodic reviews of provider software assurance process	Provider Mgmt, Provider SA Mgr	F	2.4
	6.2.5	Conduct and document periodic reviews, audits, and assessments of the development process and products	Provider SA Mgr	F	2.4

	6.2.6	Assure software problems and risks are documented and tracked to resolution	Provider SA Mgr	F	4.10.16
Software Assurance Plan	6.3	*Not a requirement*			
	6.3.1	Establish and maintain a software assurance plan	Provider Mgmt, Provider SA Mgr	F	4.3
	6.3.2	The software assurance plan shall:			
	6.3.2.1	Conform plan to IEEE 730-2002	Provider Mgmt, Provider SA Mgr	F	4.3
	6.3.2.2	Implement requirements of provider software assurance and software assurance disciplines sections into plan	Provider Mgmt, Provider SA Mgr	F	4.3
	6.3.2.3	Give precedence of software assurance Standard sections over IEEE 730-2002	Provider Mgmt, Provider SA Mgr	F	4.3
Software Assurance Plan Change Procedures	6.4	*Not a requirement*			
	6.4.1	Submit plan deviations or changes formally to acquirer	Provider SA Mgr	F	4.3
	6.4.2	Perform and submit risk analysis of deviations or changes to plan	Provider SA Mgr	F	4.3
Software Assurance Approval Authority	6.5	Have approval authority on the establishment and composition of all software baselines and any changes to the baselines	Provider SA Mgr	F	4.1
Software Assurance Records	6.6	*Not a requirement*			
	6.6.1	Prepare, maintain, and manage configuration of software assurance records	Provider SA Mgr	F	4.10.17
	6.6.2	Include recommended preventive measures, corrective actions, and lessons learned in software assurance records	Provider SA Mgr	F	1.2
Software Assurance Status Reporting	6.7	*Not a requirement*			
	6.7.1	Prepare software assurance status reports	Provider SA Mgr	F	2.4
Training	6.8	*Not a requirement*			
	6.8.1	Ensure that software assurance personnel are trained and/or experienced	Provider SA Mgr	F	4.2.3

	6.8.2	Obtain software assurance training for management, engineering, and software assurance personnel	Provider SA Mgr	F	4.2.3
	6.8.3	Ensure software assurance personnel training is current with assurance and development methods	Provider SA Mgr	F	4.2.3
	6.8.4	Ensure that software assurance personnel are trained for their assigned environment	Provider SA Mgr	F	4.2.3
	6.8.5	Ensure training records are available and maintained	Provider SA Mgr	F	4.2.3
Subcontractor Controls	6.9	*Not a requirement*		*	
	6.9.1	Flow down the requirements of this Standard to all subcontractors	Provider SA Mgr	F	2.4
	6.9.2	Assure that the subcontractors satisfy the flowed down requirements	Provider SA Mgr	F	2.4
Disciplines	7	Not a requirement			
Software Quality - Product Assurance	7.1	Not a requirement			
	7.1.1	Product assurance shall be performed to assure that:			
	7.1.1.1	All of the required plans are documented, adhere to applicable standards and procedures, are mutually consistent, and are being executed	Acquirer & Provider SA Engr	F	3.11.1 4.10.1
	7.1.1.2	All software requirements are defined, traceable from one life cycle phase to another, and analyzed	Acquirer & Provider SA Engr	F	3.11.4 4.10.4
	7.1.1.3	Evaluate software products and related documentation	Acquirer & Provider SA Engr	F	3.11.4 4.10.4
	7.1.1.4	Project documentation and any changes to them have been reviewed for impact to the quality of the product	Acquirer & Provider SA Engr	F	3.11.5 4.10.5
	7.1.1.5	Witness formal and acceptance-level software testing	Acquirer & Provider SA Engr	F	3.11.7 4.10.7
	7.1.1.6	Update, audit, and/or review lower level testing results and development folders	Acquirer & Provider SA Engr	F	3.11.7 4.10.7

	7.1.1.7	Software quality metrics are in place and are used to ensure the quality and safety of the software products	Acquirer & Provider SA Engr	F	3.11.10 4.10.10
	7.1.1.8	Specify standards and procedures for management, acquisition, engineering, and assurance activities	Acquirer & Provider SA Engr	F	3.11.8 4.10.8
	7.1.1.9	Verify software is compliant with functional and performance requirements	Acquirer & Provider SA Engr	F	3.12 4.11
	7.1.1.10	Present the status and quality of the software at formal reviews	Acquirer & Provider SA Engr/SA Mgr	F	3.12 4.11
	7.1.1.11	Report problems with software products at formal and informal reviews	Acquirer & Provider SA Engr/SA Mgr	F	3.11.16 4.10.16
Software Quality - Process Assurance	7.1.2	Process assurance shall be performed to assure that:			
	7.1.2.1	Those software life cycle processes employed for the project adhere to the applicable plans	Acquirer & Provider SA Engr	F	3.11.11 4.10.11
	7.1.2.2	Document, track, and resolve problems found with the implementation of software life cycle processes	Acquirer & Provider SA Engr/SA Mgr	F	3.11.11 4.10.11
	7.1.2.3	The software engineering practices, development environment, test environment, and libraries employed for the project adhere to applicable standards and procedures	Acquirer & Provider SA Engr	F	3.11.12 4.10.12
	7.1.2.4	Formal reviews and inspections are monitored and address software quality issues	Acquirer & Provider SA Engr	F	3.11.6 4.10.6
	7.1.2.5	Audit all management, engineering, and assurance processes for compliance with applicable plans	Acquirer & Provider SA Engr	F	3.11.11 4.10.11
	7.1.2.6	Assess the software quality metrics process for compliance to appropriate documentation or requirements	Acquirer & Provider SA Engr	F	3.11.10 4.10.10
Software Safety	7.2	Not a requirement			

	7.2.1	Implement the requirements for NASA-STD-8719.13, NASA Software Safety Standard	Acquirer & Provider	F	3.10.1 4.9.1
	7.2.2	Coordinate software safety tasks between system safety personnel and software safety personnel	Acquirer & Provider SA Mgr	F	3.10.2 4.9.2
	7.2.3	Communicate any safety risks to the appropriate safety organization	Acquirer & Provider SA Mgr	F	3.10.3 4.9.3
	7.2.4	Conduct periodic reviews and/or audits for compliance with the defined software safety process	Acquirer & Provider SA Mgr	F	3.10.7 4.9.7
Software Reliability	7.3	Not a requirement			
	7.3.1	Assure that fault tolerance and redundancy have been specified, implemented correctly, and verified by testing	Acquirer & Provider SA Engr	F	3.11.9 4.10.9
	7.3.2	Include in appropriate status reports, software reliability analyses, and measurements	Acquirer & Provider SA Engr	F	3.11.10 4.10.10
	7.3.3	Maintain the collection and classification of defects found during/from software assurance and programmatic/project formal and informal reviews	Acquirer & Provider SA Engr	F	3.11.10 4.10.10
	7.3.4	Document, monitor, analyze, and track the use of software quality metrics during each stage of development and across development and operational phases	Acquirer & Provider SA Engr	F	3.11.10 4.10.10
	7.3.5	Perform trend analyses on software quality metrics	Acquirer & Provider SA Engr	F	3.11.10 4.10.10
Software Verification and Validation	7.4	Not a requirement			
	7.4.1	Assure that software verification and validation activities occur according to established plans, policies, procedures, and standards	Acquirer & Provider SA Engr	F	3.11.13 4.10.13
	7.4.2	Participate in the formal and informal reviews	Acquirer & Provider SA Engr	F	3.11.6 4.10.6

	7.4.3	Witness or review/audit results of software testing and demonstration	Acquirer & Provider SA Engr	F	3.11.7 4.10.7
	7.4.4	Collect and use defect data to analyze software quality metrics	Acquirer & Provider SA Engr	F	3.11.10 4.10.10
	7.4.5	Collect and maintain software quality records showing the participation of software assurance staff in verification and validation efforts	Acquirer & Provider SA Engr	F	1.2
	7.4.6	Provide objective evidence to the project and NASA SMA of the software's readiness for operational release	Acquirer & Provider SA Mgr	F	3.12 4.11
Independent Verification and Validation	7.5	Not a requirement			
	7.5.1	All software projects that are identified as safety-critical or software Class A by the Software Assurance Classification Assessment shall be candidates for IV&V with safety criticality as the highest criterion	IV&V	F	3.2.1 c
	7.5.2	IV&V work shall be performed by the contractors selected and managed by the IV&V facility	IV&V	F	N/A
	7.5.3	When the IV&V function is required, the provider shall provide all required information to NASA IV&V facility personnel (This requirement includes specifying on the contracts and subcontracts, IV&V's access to system and software products and personnel)	Provider Mgmt	F	3.11.15 4.10.15
	7.5.4	The IV&V facility shall initially conduct a planning and scoping exercise to determine the specific software components to be analyzed and the tasks to be performed; the IV&V approach will be documented in an IV&V plan	IV&V	F	N/A
	7.5.5	The IV&V team shall provide input to the appropriate software assurance personnel, as well as provide feedback to the project manager as agreed in the IV&V Plan	IV&V	F	N/A

Appendix H. Software Safety Litmus Test Template

Software Safety Litmus Test Report			
1. Project Name			2. Date
3. Project Manager			
4. Software Assurance Manager			
5. Software Safety Litmus Test is applied to all projects with software to determine if the software is safety-critical[1]. The software is considered safety-critical if it meets any of the following criteria:			
Criteria	Evaluation		
	Yes	No	Comments
a. Resides in a safety-critical system (as determined by a hazard analysis) AND at least one of the following apply:	☐	☐	
(1) Causes or contributes to a hazard.	☐	☐	
(2) Provides control or mitigation for hazards.	☐	☐	
(3) Controls safety-critical functions.	☐	☐	
(4) Processes safety-critical commands or data[2].	☐	☐	
(5) Detects and reports, or takes corrective action, if the system reaches a specific hazardous state.	☐	☐	
(6) Mitigates damage if a hazard occurs.	☐	☐	
(7) Resides on the same system (processor) as safety-critical software[3].	☐	☐	
b. Processes data or analyzes trends that lead directly to safety decisions (e.g., determining when to turn power off to a wind tunnel to prevent system destruction).	☐	☐	
c. Provides full or partial verification or validation of safety-critical systems, including hardware or software subsystems.	☐	☐	

[1] NASA defines safety criticality from the definition of hazard severity in NPR 8715.3, NASA Safety Manual, Chapter 3, System Safety, and Appendix D Analysis Techniques.

[2] If data is used to make safety decisions (either by a human or the system), then the data is safety-critical, as is all the software that acquires, processes, and transmits the data. However, data that may provide safety information but is *not* required for safety or hazard control (such as engineering telemetry) is not safety-critical.

[3] Non-safety-critial software residing with safety-critical software is a concern because it may fail in such a way as to disable or impair the functioning of the safety-critical software. Methods to separate the code, such as portioning, can be used to limit the software defined as safety-critical. If such methods are used, then the isolation method is safety-critical, but the isolated non-critical code is not.

Appendix I. Software Assurance Classification Report Template

Software Assurance Classification Report								
1. Project Name							2. Date	
3. Project Manager								
4. Software Assurance Manager								
Software Assurance Classification Criteria								
5a. Software Safety Litmus Test Is the Software Safety-Critical? Is Human Life a Risk Factor?					Yes ☐ ☐		No ☐ ☐	
5b. Determination for Class E, F, G, or H Software If F or G, is SA being performed? OSMA Involvement?					Yes ☐ ☐ ☐		No ☐ ☐ ☐	
5c. Software Classification Score					Score:			
5d. Software Class	A ☐	B ☐	C ☐	D ☐	E ☐	F ☐	G ☐	H ☐
5e. Software Assurance Effort/Priority	Full/ High ☐	Full/ Medium -High ☐	Medium / Medium ☐	Minimal/ Low ☐	None ☐			
6. Comments								
7. Date	Signature of Software Assurance Manager							
8. Date	Signature of Project Manager							

Appendix J. References

a. NPR 2810.1, “Security of Information Technology”

b. GLPR 8700.4, “Product Assurance”

c. GLPR 8730.5, “GRC Business Management System Quality System Manual”

d. NASA-STD-2202-93, “NASA Software Formal Inspection Standard”

e. NASA-GB-A201, “NASA Software Assurance Guidebook”

f. NASA-GB-A301, “NASA Software Quality Assurance Audits Guidebook”

g. NASA-GB-8719.13, “NASA Software Safety Guidebook”